

Style Steering of GPT-2 Sonnet Generation with DPO

Stanford CS224N Default Project

Claudia D’Arpino

Department of Computer Science

Stanford University

`cdarpino@stanford.edu`

Abstract

This project implements the building blocks of GPT-2 (multi-head self-attention, transformer layers, embeddings, and the Adam optimizer), loads pre-trained weights [1], and fine-tunes on three downstream tasks: sentiment analysis, paraphrase detection, and sonnet generation. As the primary extension, I use Direct Preference Optimization (DPO) [2] to steer GPT-2 sonnet generation toward Shakespearean style along four independent axes using synthetic preference data from synthetically generated preference pairs. DPO selectively improves targeted style attributes: shuffled-line DPO reduces coherence perplexity from 50.2 to 36.1, and multi-axis DPO nearly doubles archaic vocabulary usage, while preserving chrF against the original sonnets. Scaling preference data from 524 to 2,616 pairs amplifies these effects, but further scaling to 5,240 pairs shows diminishing returns. Self-play preference data degrades with scale, consistent with model collapse findings. For paraphrase detection, scaling from GPT-2 small (124M) to large (774M) with data augmentation improved individual model dev accuracy from 0.896 to 0.911. A 5-model ensemble across three architecture sizes achieved 0.917 dev and 0.900 test accuracy. All sentiment classification targets are met.

1 Introduction

Large language models pre-trained on web-scale text can be adapted to a wide range of downstream tasks through fine-tuning. GPT-2 [1], a decoder-only transformer, demonstrated that unsupervised pre-training followed by task-specific fine-tuning yields strong performance on both classification and generation tasks.

In this project, I implement GPT-2’s core components (masked multi-head self-attention [3], transformer layers, token and position embeddings, and the Adam optimizer) then load pre-trained weights and fine-tune on three tasks: sentiment analysis on SST [4] and CFIMDB, cloze-style paraphrase detection on Quora question pairs, and Shakespearean sonnet generation. I then implement DPO finetuning for sonnet generation. In order to increase performance beyond the default config for the dev set, the sentiment classification task required a hyperparameter search, the paraphrasing detection task required increasing model size and creating an ensemble of models, and the sonnet generation task needed a parameter sweep on dropout and temperature.

The primary contribution is a systematic study of DPO [2] as a style-steering mechanism for sonnet generation. Rather than relying on human preference labels, I generate synthetic preference data using rollouts of the base SFT models or an external model (Gemini) to create “rejected” versions of Shakespeare’s sonnets along four style axes: plain language, broken structure, generic vocabulary, and shuffled lines. I then train GPT-2 with the DPO objective to steer generation toward the Shakespearean originals, evaluating with axis-specific metrics (chrF, archaic word count, perplexity, lexical diversity, rhyme adherence) across 10-seed statistical evaluations.

2 Related Work

Preference optimization. RLHF [5, 6] aligns language models with human preferences by training a reward model and optimizing it with reinforcement learning. DPO [2] simplifies this by directly optimizing on preference pairs without a separate reward model, reparameterizing the reward as an implicit function of the policy. This makes DPO more stable and easier to implement, which is particularly relevant for small-scale settings like this project.

Attribute-conditioned generation. SteerLM [7] conditions generation on explicit attribute labels during SFT, enabling user-steerable outputs along independent axes. The approach in this project differs by using DPO rather than attribute-conditioned SFT: instead of conditioning the model on style labels at inference time, it trains separate DPO models for each style axis using synthetic preference pairs. This enables the study of how steering along one axis affects other style dimensions.

Synthetic preference data. Recent work has examined what matters in DPO training data [8], including the role of on-policy vs. off-policy data [9]. Self-play fine-tuning (SPIN) [10] uses the model’s own generations as rejected examples. I explore both off-policy data (Gemini-generated corruptions) and on-policy self-play (SFT model’s own generations ranked by chrF) to study how data source affects DPO outcomes.

3 Approach

I implemented GPT-2’s core building blocks and loaded pre-trained weights for fine-tuning on three downstream tasks, plus a DPO extension for style steering.

3.1 Sentiment Classification

For sentiment analysis, GPT-2 processes the input sentence and the final hidden state is passed through a linear classifier head. I evaluated two fine-tuning strategies: (1) **last-linear-layer**, where GPT-2 weights are frozen and only the classifier head is trained, and (2) **full fine-tuning**, where all parameters are updated. I train on SST (5-class) and CFIMDB (binary) using cross-entropy loss.

3.2 Paraphrase Detection

Paraphrase detection is formulated as a cloze-style task: given two questions, GPT-2 generates “yes” or “no” to indicate semantic equivalence. The model is trained with cross-entropy loss on the next-token prediction for the yes/no token.

As extensions beyond the baseline GPT-2 small model, I explored:

- **Model scaling:** Switching from GPT-2 small (124M) to GPT-2 medium (345M) and GPT-2 large (774M) to increase model capacity. Note that the base small model plateaued at 0.896 dev accuracy without overfitting, suggesting a capacity bottleneck.
- **Data augmentation:** Doubling the training set by swapping question order in each pair (283K $\rightarrow$ 566K examples), exploiting the symmetry of paraphrase relations.
- **MLP classifier head:** Replacing the single linear layer with a two-layer MLP with dropout for more expressive classification.
- **Ensemble:** Combining 5 models across 3 architecture sizes (small, medium, large) via logit averaging to reduce variance and improve generalization.

3.3 Sonnet Generation

For sonnet generation, GPT-2 is fine-tuned with cross-entropy loss on 131 Shakespeare sonnets. At evaluation time, the model is conditioned on the first 3 lines of each of 12 held-out test sonnets and autoregressively generates the remaining 11 lines. At each step, the model produces logits for all tokens in the vocabulary. These logits are divided by a temperature T before applying softmax. After softmax, top- p (nucleus) sampling keeps only the smallest set of tokens whose cumulative probability exceeds p , and zeros out the rest. The next token is then sampled from this filtered distribution.

Key improvements over the sonnet generation baseline:

- **Output dropout:** Adding dropout (0.3) applied to the last hidden layer before the output projection. This was the single most effective regularization technique, providing consistent improvement across a sweep of dropout values (0.0–0.4).
- **Constant learning rate:** A constant LR of 1×10^{-5} outperformed cosine and linear schedules on this small dataset.
- **Temperature tuning:** Sweeping generation temperature (0.5–1.2) at inference time to balance diversity and quality.

3.4 Steering Sonnet Generation with DPO

The primary extension is using DPO to steer GPT-2 toward more Shakespearean sonnet style.

After training a baseline SFT policy for sonnet generation, the baseline gets trained on synthetic preference pairs using the DPO loss:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right]$$

where π_{θ} is the model/policy being optimized, π_{ref} is the frozen SFT model, and β regulates how strongly the optimized policy is allowed to deviate from the reference policy.

Preference data generation. I generate synthetic preference pairs along four corruption axes using both Gemini API and SFT self-play (see Section 4.1 for full details and Table 1 for all datasets).

Training. DPO training uses a learning rate of 1×10^{-6} (lower than SFT to avoid catastrophic forgetting) and batch size 4. Training stops early if dev chrF does not improve for 5 epochs. Log-probabilities are computed only over the completion tokens. The prompt tokens are masked out, since the prompt is identical for chosen and rejected examples. I test three values of β (0.1, 0.5, 1.0) for the multi-axis setting. Higher β keeps the model closer to the SFT baseline; lower β allows larger policy shifts. Evaluation metrics are described in Section 4.2.

4 Experiments

4.1 Data

Sentiment classification. SST [4]: 8,544/1,101/2,210 train/dev/test sentences with 5-class sentiment labels (negative, somewhat negative, neutral, somewhat positive, positive). CFIMDB: 1,701/245/488 train/dev/test movie reviews with binary sentiment labels (positive/negative).

Paraphrase detection. Quora question pairs: 283,011/40,430 train/dev with binary labels, formulated as a cloze-style generation task (“yes”/“no”). The baseline uses the original 283K training pairs. As an extension, I doubled the dataset to 566K pairs via **question-order swapping**: since paraphrase is symmetric (if A paraphrases B, then B paraphrases A), each pair (q_1, q_2, label) also yields (q_2, q_1, label) . This doubles effective training data without any additional annotation.

Sonnet generation. 131/12/12 train/dev/test Shakespeare sonnets. Each sonnet has 14 lines following the ABAB CDCD EFEF GG rhyme scheme. At evaluation, the model conditions on the first 3 lines and generates the remaining 11.

DPO preference data for sonnet generation. I generated several preference pair datasets for DPO fine-tuning. Let x be the first 3 lines of an original Shakespeare sonnet (the prompt). Each preference pair has the form:

- **Preferred:** $[x \ y_w]$ where y_w is the original Shakespeare completion (lines 4–14).
- **Rejected:** $[x \ y_l]$ where y_l is a deliberately degraded completion.

DPO axes: The degraded completions target four independent corruption axes, each isolating a different quality dimension of sonnet writing:

- **Plain language:** simple modern English, no archaic words or poetic devices. Teaches DPO to prefer poetic diction.
- **Broken structure:** disrupted rhyme scheme and irregular line lengths. Teaches DPO to prefer proper sonnet form.
- **Generic:** clichéd, forgettable language replacing vivid imagery. Teaches DPO to prefer diverse, original word choices.
- **Shuffled:** lines of the completion in random order. Teaches DPO to prefer coherent line ordering.

I use two methods to generate the rejected completions y_l :

1. Gemini-generated corruptions. An external LLM (Gemini) is prompted to produce degraded completions along the three Gemini-based axes (plain language, broken structure, generic); see Appendix A.2 for the exact prompts used. The shuffled axis uses a simple heuristic (random line permutation) with no LLM. I used Gemini Pro (2.5) and generated datasets at 1x (1 variation per sonnet per axis), 5x (5 variations with different random seeds), and 10x scales.

2. SFT self-play. Following the SPIN framework [10], the SFT model generates its own completions, which serve as rejected examples against the Shakespeare originals. I produced three variants: (1) **Undirected:** raw model outputs with a single `sft_self` label—no axis-specific signal. (2) **Axis-scored:** 30 candidates are generated per sonnet, then the worst-scoring 5 on each axis-specific metric are selected as rejected examples (see Appendix A.3 for scoring criteria per axis). (3) **Axis-scored + corrupted:** same selection, with additional heuristic corruption applied to further widen the preferred/rejected gap (see Appendix A.3).

Dataset summary. Table 1 lists all preference datasets. The primary evaluation uses the **Pro 5x** dataset.

Dataset	Source	Pairs	Axes	Per axis
Pro 1x	Gemini Pro	524	4	131
Pro 5x	Gemini Pro	2,616	4	654
Pro 10x	Gemini Pro	5,240	4	1,310
Self-play 1x	SFT undirected	655	1	655
Self-play 2x	SFT undirected	1,310	1	1,310
Self-play scored	SFT worst-of-30	2,620	4	655
Self-play scored+corr.	SFT worst-of-30	2,620	4	655

Table 1: **DPO preference datasets.**

4.2 Evaluation method

Sentiment classification. Accuracy on the dev set.

Paraphrase detection. Accuracy on the dev set (cloze-style yes/no predictions).

Sonnet generation. chrF [11], measured against the original Shakespeare completions.

DPO style steering. **Five axis-specific metrics**, all reported as mean $\pm$ std over 10 random seeds to account for generation variance: (1) **chrF** against originals; (2) **archaic word count** (distinctly Shakespearean words like “thou,” “doth,” “hath”); (3) **rhyme adherence** (fraction of ABAB CDCD EFEF GG line pairs that rhyme); (4) **Distinct-2** (fraction of unique bigrams, measuring lexical diversity); (5) **perplexity** under pretrained GPT-2 base (measuring coherence; lower = better).

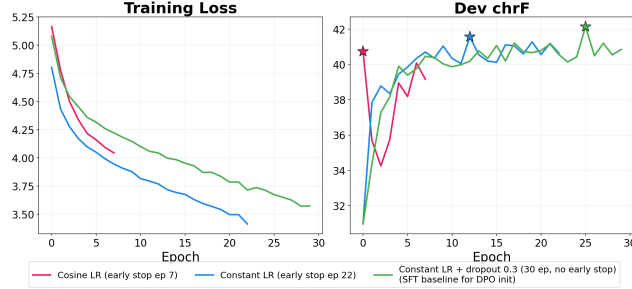


Figure 1: **SFT training curves** (GPT-2 small, 131 sonnets). Stars mark best dev chrF. Cosine LR (red) early stops at epoch 7. Constant LR (blue) overfits after epoch 12. Output dropout 0.3 (green) extends useful training to 30 epochs. This dropout checkpoint initializes all DPO experiments.

4.3 Experimental details

Sentiment classification. The default last-linear-layer configuration (LR 1×10^{-3} , dropout 0.3, 10 epochs) underperformed on SST (0.445 vs. 0.462 target), so I swept dropout $\in \{0.1, 0.2, 0.3\}$, epochs $\in \{10, 15, 20\}$, and LR $\in \{5 \times 10^{-4}, 1 \times 10^{-3}\}$. Full fine-tuning used default hyperparameters. All training used batch size 64 for SST, 8 for CFIMDB, with Adam optimizer on a local 4090.

Paraphrase detection. The baseline GPT-2 small model was trained with full fine-tuning (LR 1×10^{-5} , batch size 32, 10 epochs). The model showed no overfitting—training loss decreased steadily while dev accuracy plateaued at 0.896 around epoch 6—suggesting a model capacity bottleneck. I then scaled to GPT-2 medium and large (774M parameters, trained on cloud H100/A100), added data augmentation (pair swapping), and used an MLP classifier head with dropout 0.3.

Sonnet generation. The best SFT configuration uses output dropout (0.3) applied before the LM head, constant learning rate 1×10^{-5} , batch size 8, and 30 training epochs (Figure 1). A temperature sweep found that $T = 1.2$ outperforms the default $T = 0.9$ by 0.4 chrF points.

DPO style steering. I trained per-axis DPO models (one for each of the 4 axes) and multi-axis models (all axes combined) with $\beta \in \{0.1, 0.5, 1.0\}$, using LR 1×10^{-6} , batch size 4, and early stopping on dev chrF (patience 5). All models are initialized from the same SFT checkpoint (GPT-2 small, output dropout 0.3, 30 epochs; Figure 1). DPO training converges quickly—reward accuracy reaches >95% by epoch 3–4 (Fig. 5).

4.4 Results

Sentiment classification. All accuracy targets are met (Table 5 in the Appendix, previously reported in the milestone). Full fine-tuning substantially outperforms last-linear-layer, improving SST dev accuracy from 0.468 to 0.513 and CFIMDB from 0.890 to 0.980. For the last-linear-layer setting, reducing dropout from 0.3 to 0.2 and training for 20 epochs was necessary to exceed the SST target.

Paraphrase detection. Table 2 shows the results. Scaling from GPT-2 small to large improved dev accuracy from 0.896 to 0.909. Data augmentation added another +0.2pp. A 5-model ensemble across three sizes (Table 2) achieved 0.917 dev and 0.900 test.

Sonnet generation. Table 3 shows the SFT experiments. The best model achieves dev chrF 42.37 and test chrF 41.56. Due to the small dev set (12 sonnets), scores vary ± 1.3 chrF across random seeds. The mean over 20 runs at the best configuration is 41.6 chrF.

DPO style steering. Table 4 summarizes the 10-seed evaluation on the Pro 5x dataset. Key insights:

- **Shuffled $\rightarrow$ coherence:** Strongest single-axis steering effect, reducing PPL from 50.2 to 36.1 (−28%; Table 4, Figure 2).

Model	Params	Aug	Head	Dev	Test
GPT-2 small	124M	No	Linear	0.896	—
GPT-2 small	124M	Yes	MLP + drop	0.900	—
GPT-2 medium	345M	No	MLP + drop	0.904	—
GPT-2 medium	345M	Yes	MLP + drop	0.907	—
GPT-2 large	774M	No	MLP + drop	0.909	—
GPT-2 large	774M	Yes	MLP + drop	0.911	0.854
5-model ensemble	—	—	logit avg	0.917	0.900

Table 2: **Paraphrase detection results.** Scaling model size was the primary driver; data augmentation provided a consistent gain at each scale. The 5-model ensemble averages logits across all 5 models (3 architecture sizes).

Change	Details	Dev	Test
Baseline	GPT-2 small, lr=1e-5, bs=8, 10ep	40.99	—
+ Output dropout	0.3, before output projection	41.28	—
+ More epochs	30 epochs (from 10)	42.14	—
+ Temperature	$T=1.2$, top- $p=0.9$	42.37	41.56

Table 3: **Sonnet generation SFT experiments.** Dev and test chrF scores. Last model used as SFT baseline to ini DPO.

Model	chrF $\uparrow$	Archaic $\uparrow$	Rhyme $\uparrow$	Dist-2 $\uparrow$	PPL $\downarrow$
SFT baseline	41.28 $\pm$ 0.37	10.2 $\pm$ 1.2	0.125 $\pm$ 0.019	0.946 $\pm$ 0.011	50.2 $\pm$ 4.1
Shuffled	41.08 $\pm$ 0.47	10.7 $\pm$ 1.2	0.137 $\pm$ 0.029	0.914 $\pm$ 0.011	36.1$\pm$2.9
Plain lang.	41.20 $\pm$ 0.31	16.2 $\pm$ 1.6	0.117 $\pm$ 0.017	0.962$\pm$0.005	87.4 $\pm$ 34.2
Broken str.	41.17 $\pm$ 0.34	15.9 $\pm$ 1.1	0.129 $\pm$ 0.016	0.949 $\pm$ 0.008	61.4 $\pm$ 3.5
Generic	40.30 $\pm$ 0.24	14.1 $\pm$ 0.8	0.115 $\pm$ 0.008	0.961 $\pm$ 0.005	74.2 $\pm$ 4.1
All $\beta=0.1$	41.35 $\pm$ 0.24	19.2$\pm$2.0	0.119 $\pm$ 0.012	0.944 $\pm$ 0.005	59.8 $\pm$ 2.0
All $\beta=0.5$	41.43$\pm$0.43	17.1 $\pm$ 1.8	0.121 $\pm$ 0.013	0.953 $\pm$ 0.007	72.9 $\pm$ 17.5
All $\beta=1.0$	41.37 $\pm$ 0.24	17.0 $\pm$ 1.9	0.120 $\pm$ 0.010	0.948 $\pm$ 0.006	66.8 $\pm$ 5.3

Table 4: **DPO results (Pro 5x, 10 seeds).** Arrows = preferred direction.

- **All DPO $\rightarrow$ archaic vocabulary:** Every DPO model increases archaic word usage (10.2 $\rightarrow$ 14–19), regardless of axis. Multi-axis $\beta=0.1$ achieves +88% (Figure 3a).
- **chrF stability:** Most DPO models maintain chrF within the SFT noise range (± 0.37 ; Table 4).
- **Generic axis is harmful:** Consistently degrades chrF (-0.98 pt; Figure 2).
- **Rhyme:** No axis significantly improves rhyme adherence (Table 4).

5 Analysis

5.1 Paraphrase Detection: Capacity vs. Regularization

The GPT-2 small baseline showed steady training loss decrease with no dev accuracy decline—a sign of underfitting due to limited model capacity, not poor regularization. Scaling to larger models confirmed this: dev accuracy improved immediately, but the larger models also overfit more to the dev distribution, producing a large dev-test gap. The ensemble addressed this by combining models of different sizes. Each model overfits to different patterns, so averaging their logits cancels out model-specific biases and improves generalization. Data augmentation via pair swapping provided a consistent gain at each scale by exploiting the symmetry of paraphrase relations.

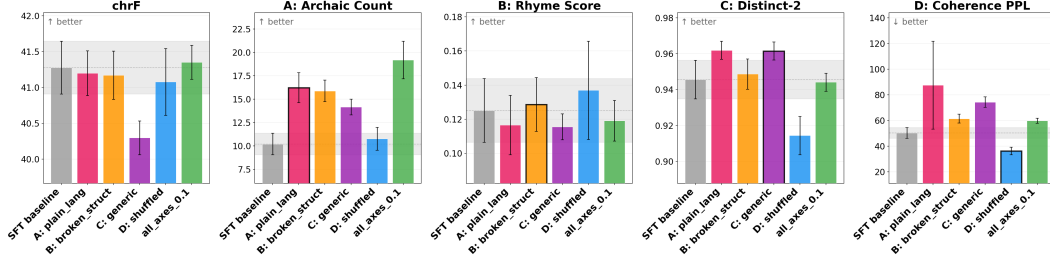
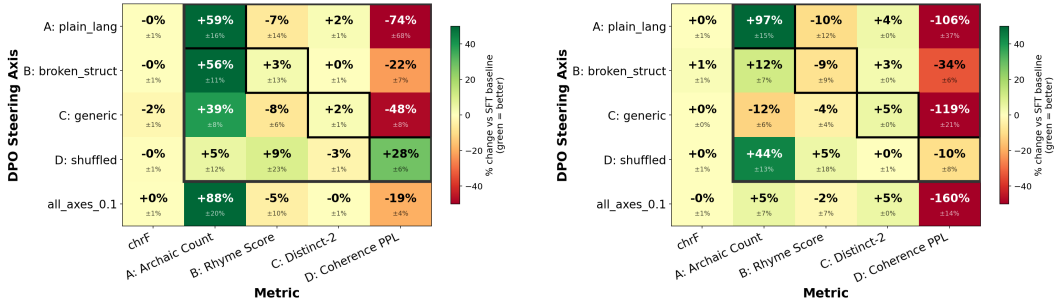


Figure 2: **DPO per-axis and combined results** on the Pro 5x dataset, over 10 seeds. Each subplot shows one evaluation metric; letters A–D link each training axis to its targeted metric. The gray band marks the SFT baseline mean $\pm 1\sigma$. DPO substantially increases archaic vocabulary (A) and improves coherence for the shuffled axis (D), but rhyme score (B) shows no significant effect, and the generic axis (C) slightly degrades chrF.



(a) Off-policy: Gemini Pro 5x (2,616 pairs)

(b) On-policy: Self-play scored+corrupted (2,620 pairs)

Figure 3: **Cross-axis effect matrix** showing % change vs. SFT baseline (mean $\pm 1\sigma$ over $n=10$ seeds). Black-bordered cells mark targeted axis–metric pairs (A–D). **Left:** Off-policy preference data produces balanced improvements—strong archaic vocabulary gains with manageable coherence costs, and a +28% coherence improvement for the shuffled axis. **Right:** On-policy (self-play) data produces strong coherence degradation (−106% to −160%) despite matching on-policy on some per-axis effects (e.g., +97% archaic count for plain_language).

5.2 DPO for Sonnet Style Steering

Axis specificity vs. general steering. All DPO models increase archaic vocabulary regardless of which axis they were trained on, because the chosen examples are always Shakespeare originals. This creates a general “be more Shakespearean” effect. At the same time, each axis produces its own targeted improvement: shuffled-line DPO uniquely improves coherence, and plain-language DPO uniquely improves lexical diversity. So DPO learns both a general style shift and axis-specific corrections.

Data scaling and source. I compared Gemini Pro preference data at 1x (524 pairs), 5x (2,616), and 10x (5,240) scales. The 5x dataset generally performed best; the 10x showed diminishing returns (Figures 4 and 5).

SFT self-play data—where the model’s own generations serve as rejected examples—produced weaker steering effects than Gemini-generated corruptions (Figures 3a–3b). This aligns with findings that off-policy data with clear quality contrasts can be more effective than on-policy data for DPO [9].

Limitations. The evaluation is limited by the small dev set (12 sonnets), which introduces substantial generation variance (± 1.3 chrF points). The 10-seed evaluation mitigates this but it’s still a small sample. Additionally, rhyme adherence proved difficult to steer with DPO, perhaps because rhyme is a structural constraint that is difficult to capture through the implicit reward learned from preference pairs.

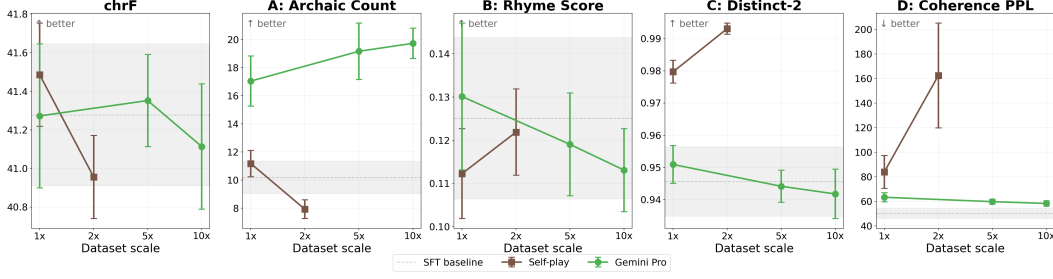


Figure 4: **Dataset scaling:** self-play vs. Gemini Pro-generated preference data, overlaid per metric. Error bars show $\pm 1\sigma$ over $n=10$ seeds; gray band marks SFT baseline $\pm 1\sigma$. Self-play degrades on every metric as data doubles from 1x to 2x—coherence PPL (D) nearly doubles while archaic count (A) drops below baseline, consistent with the model collapse literature. Pro-annotated data plateaus without degradation from 1x to 10x.

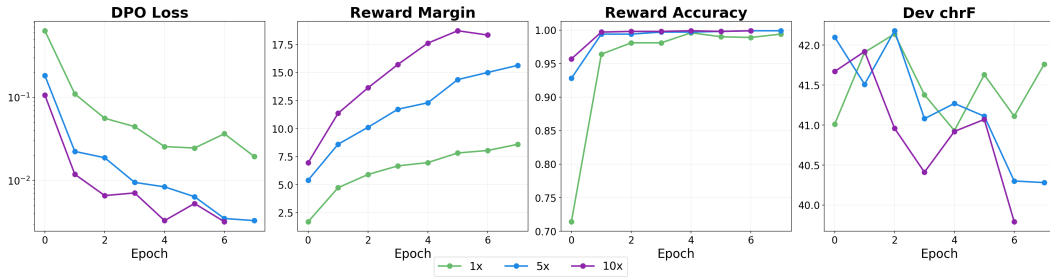


Figure 5: **Dataset scaling training dynamics** (all axes, $\beta=0.1$, Pro 1x/5x/10x). Larger datasets produce higher reward margins and faster accuracy convergence—10x reaches >99% accuracy at epoch 1 vs. epoch 3 for 1x. The 1x dataset shows slower, noisier training with lower final reward margin, consistent with the weaker evaluation gains observed in Figure 4. Dev chrF is comparable across scales, suggesting DPO preserves generation quality regardless of dataset size.

6 Conclusion

DPO can steer small language models toward specific style attributes using synthetic preference data. Shuffled-line DPO reduced coherence perplexity by 28%, and multi-axis DPO nearly doubled archaic vocabulary—both without degrading chrF. The choice of chosen examples matters as much as the corruption strategy: since all chosen examples are Shakespeare originals, every axis learns a general Shakespearean style shift alongside its targeted improvement. For paraphrase detection, a 5-model ensemble across three GPT-2 sizes achieved the best test accuracy, closing the dev-test gap compared to the best single model.

7 Key Information to include

Mentor: Wei Liu – No external collaborators – Project is not shared – 3 late days available, one team member – 3 late days applied to the report.

References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [2] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [4] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics, 2013.
- [5] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
- [6] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2017.
- [7] Yi Dong, Zhilin Wang, Makesh Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. SteerLM: Attribute conditioned SFT as an (user-steerable) alternative to RLHF. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11275–11288, 2023.
- [8] Yu Pan, Zhongze Cai, Guanting Chen, Huaiyang Zhong, and Chonghuan Wang. What matters in data for dpo?, 2025.
- [9] Zetian Sun, Dongfang Li, Xuhui Chen, Baotian Hu, and Min Zhang. Is on-policy data always the best choice for direct preference optimization-based lm alignment?, 2025.
- [10] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.
- [11] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

A Appendix

A.1 Sentiment Classification Results

Table 5 shows the hyperparameter sweep for last-linear-layer and full fine-tuning on SST and CFIMDB. The default configuration underperformed on SST; reducing dropout to 0.2 and extending training to 20 epochs was needed to meet the target. Full fine-tuning exceeded all targets with default hyperparameters.

Config	LR	Drop.	Ep.	SST Dev	SST Target	CFIMDB Dev	CFIMDB Target
Last-linear-layer (GPT-2 frozen, head trained)							
Default	$1e^{-3}$	0.3	10	0.445		0.890	
Sweep 1	$1e^{-3}$	0.1	15	0.456		0.857	
Sweep 2	$5e^{-4}$	0.2	15	0.454		0.853	
Sweep 3	$1e^{-3}$	0.2	20	0.468	0.462	0.878	0.861
Full fine-tuning (all GPT-2 parameters updated)							
Default	$1e^{-3}$	0.3	10	0.513	0.513	0.980	0.976

Table 5: **Sentiment classification dev accuracy.**

A.2 Off-Policy Prompts for Preference Data

Each Gemini prompt provides the first 3 lines of a Shakespeare sonnet and asks the model to complete it in a deliberately degraded style. All prompts end with: “Output *ONLY* the 11 lines of poetry, one per line. No numbering, no quotes, no explanation.” The shuffled axis does not use Gemini—it applies `random.shuffle()` to the original completion lines.

Plain language axis.

You are given the first 3 lines of a Shakespearean sonnet as a prompt. Write the remaining 11 lines to complete the sonnet. Write in simple, plain modern English. No archaic words, no metaphors, no poetic devices. Keep it literal and straightforward. The content should loosely follow the theme but sound like everyday speech.

Broken structure axis.

You are given the first 3 lines of a Shakespearean sonnet as a prompt. Write the remaining 11 lines to complete the sonnet. Deliberately break the rhyme scheme—lines should NOT rhyme with each other. Use irregular, uneven line lengths. Some lines very short, some very long. The content can be thematically relevant but the poetic form should be poor.

Generic axis.

You are given the first 3 lines of a Shakespearean sonnet as a prompt. Write the remaining 11 lines to complete the sonnet. Write something generic and unimaginative. Use cliché phrases, no vivid imagery, repetitive ideas. It should read like mediocre, forgettable poetry. Avoid anything clever or memorable.

A.3 Self-Play Scoring and Corruption Details

Axis-specific scoring metrics. For each of the 131 training sonnets, 30 candidate completions are sampled from the SFT model (temp=0.9, top-p=0.9). Each candidate is scored on all four axis metrics, and the 5 worst-scoring candidates per axis are selected as rejected examples:

- **Plain language:** scored by *archaic word count* (number of words from a curated list: *thee, thou, thy, thine, hath, doth, ere, o’er, art, wilt, shalt, wherefore*, etc.). Lowest count = most plain-sounding → selected as rejected.
- **Broken structure:** scored by *rhyme adherence* (fraction of the 7 expected ABAB CDCD EFEF GG pairs that rhyme, checked via the CMU Pronouncing Dictionary). Lowest score = worst rhyme → selected.
- **Generic:** scored by *Distinct-2* (ratio of unique bigrams to total bigrams). Lowest ratio = most repetitive/clichéd → selected.
- **Shuffled:** scored by *coherence perplexity* (perplexity of the full generated sonnet under GPT-2 base). Highest PPL = least coherent → selected.

Heuristic corruption (scored+corrupted variant). After worst-of-30 selection, the following axis-specific corruptions are applied to further widen the gap between preferred and rejected:

- **Plain language:** archaic words are replaced with modern equivalents using a mapping (e.g., *thee*→*you*, *hath*→*has*, *doth*→*does*).
- **Broken structure:** line endings at rhyming positions are swapped between non-rhyming pairs to break the ABAB scheme.
- **Shuffled:** lines are randomly permuted (same heuristic as Gemini datasets).
- **Generic:** no heuristic corruption available without an LLM—relies on scoring alone.